

# *OpenCMISS-iron* examples and tests used by *OpenCMISS* developers at University of Stuttgart, Germany

Christian Bleiler\*, Andreas Hessenthaler\*,  
Thomas Klotz\*, Aaron Krämer†, Benjamin Maier‡,  
Sergio Morales\*, Mylena Mordhorst\*, Harry Saini\*

June 30, 2017  
10:16

## CONTENTS

1	Introduction	4
1.1	Cmgui files for cmgui-2.9 . . . . .	4
1.2	Variations to consider . . . . .	4
1.3	Folder structure . . . . .	5
2	How to work on this document	5
3	Diffusion equation	6
3.1	Equation in general form . . . . .	6
3.2	Example-0001 . . . . .	7
3.2.1	Mathematical model - 2D . . . . .	7
3.2.2	Mathematical model - 3D . . . . .	7
3.2.3	Computational model . . . . .	7
3.2.4	Result summary . . . . .	8
3.3	Example-0001-u . . . . .	11
3.3.1	Mathematical model - 2D . . . . .	11
3.3.2	Mathematical model - 3D . . . . .	11
3.3.3	Computational model . . . . .	11
3.3.4	Result summary . . . . .	12
3.4	Example-0002 . . . . .	15
3.4.1	Mathematical model - 2D . . . . .	15
3.4.2	Mathematical model - 3D . . . . .	15
3.4.3	Computational model . . . . .	15
3.4.4	Result summary . . . . .	16
3.5	Example-0011 . . . . .	19

\* Institute of Applied Mechanics (CE), University of Stuttgart, Pfaffenwaldring 7, 70569 Stuttgart, Germany

† Institute for Parallel and Distributed Systems, University of Stuttgart, Universitätsstraße 38, 70569 Stuttgart, Germany

‡ Lehrstuhl Mathematische Methoden für komplexe Simulation der Naturwissenschaft und Technik, University of Stuttgart, Allmandring 5b, 70569 Stuttgart, Germany

3.5.1	Mathematical model - 2D . . . . .	19
3.5.2	Mathematical model - 3D . . . . .	19
3.5.3	Computational model . . . . .	19
3.5.4	Result summary . . . . .	20
4	Linear elasticity . . . . .	23
4.1	Equation in general form . . . . .	23
4.2	Example-0101 . . . . .	24
4.2.1	Mathematical model . . . . .	24
4.2.2	Computational model . . . . .	24
4.2.3	Results . . . . .	24
4.2.4	Validation . . . . .	24
5	Finite elasticity . . . . .	27
6	Navier-Stokes flow . . . . .	28
7	Monodomain . . . . .	29
8	CellML model . . . . .	30

## LIST OF FIGURES

Figure 1	2D results, iron reference w/ command line arguments [2.0 1.0 0.0 8 4 0 1 0]. . . . .	8
Figure 2	2D results, current run w/ command line arguments [2.0 1.0 0.0 8 4 0 1 0]. . . . .	9
Figure 3	3D results, iron reference w/ command line arguments [2.0 1.0 1.0 8 4 4 1 0]. . . . .	9
Figure 4	3D results, current run w/ command line arguments [2.0 1.0 1.0 8 4 4 1 0]. . . . .	10
Figure 5	2D results, iron reference w/ command line arguments [2.0 1.0 0.0 8 4 0 1 0]. . . . .	12
Figure 6	2D results, current run w/ command line arguments [2.0 1.0 0.0 8 4 0 1 0]. . . . .	13
Figure 7	3D results, iron reference w/ command line arguments [2.0 1.0 1.0 8 4 4 1 0]. . . . .	13
Figure 8	3D results, current run w/ command line arguments [2.0 1.0 1.0 8 4 4 1 0]. . . . .	14
Figure 9	2D results, iron reference w/ command line arguments [2.0 1.0 0.0 8 4 0 1 0]. . . . .	16
Figure 10	2D results, current run w/ command line arguments [2.0 1.0 0.0 8 4 0 1 0]. . . . .	17
Figure 11	3D results, iron reference w/ command line arguments [2.0 1.0 1.0 8 4 4 1 0]. . . . .	17
Figure 12	3D results, current run w/ command line arguments [2.0 1.0 1.0 8 4 4 1 0]. . . . .	18
Figure 13	2D results, iron reference w/ command line arguments [2.0 1.0 0.0 8 4 0 1 0 1 1]. . . . .	21
Figure 14	2D results, current run w/ command line arguments [2.0 1.0 0.0 8 4 0 1 0 1 1]. . . . .	21
Figure 15	3D results, iron reference w/ command line arguments [2.0 1.0 1.0 8 4 4 1 0 1 1 1]. . . . .	22
Figure 16	3D results, current run w/ command line arguments [2.0 1.0 1.0 8 4 4 1 0 1 1 1]. . . . .	22
Figure 17	Results, analytical solution. . . . .	24

Figure 18	Results, Abaqus reference. . . . .	25
Figure 19	Results, iron reference. . . . .	25
Figure 20	Results, current run. . . . .	26

## LIST OF TABLES

Table 1	Initials of people working on examples, in alphabetical order (surnames). . . . .	5
---------	---	---

## 1 INTRODUCTION

This document contains information about examples used for testing *OpenCMISS-iron*. Read: How-to<sup>1</sup> and [1].

### 1.1 Cmgui files for cmgui-2.9

### 1.2 Variations to consider

- Geometry and topology
  - 1D, 2D, 3D
  - Length, width, height
  - Number of elements
  - Interpolation order
  - Generated or user meshes
  - quad/hex or tri/tet meshes
- Initial conditions
- Load cases
  - Dirichlet BC
  - Neumann BC
  - Volume force
  - Mix of previous items
- Sources, sinks
- Time dependence
  - Static
  - Quasi-static
  - Dynamic
- Material laws
  - Linear
  - Nonlinear (Mooney-Rivlin, Neo-Hookean, Ogden, etc.)
  - Active (Stress, strain)
- Material parameters, anisotropy
- Solver
  - Direct
  - Iterative
- Test cases
  - Numerical reference data
  - Analytical solution
- A mix of previous items

---

<sup>1</sup> <https://bitbucket.org/hessenthaler/opencmisshowto>

### 1.3 Folder structure

TBD..

## 2 HOW TO WORK ON THIS DOCUMENT

In the Google Doc at [https://docs.google.com/spreadsheets/d/1RGKj8vVPqQ-PH0UwMX\\_e9TAzqaYavKi0z0D4pKY9RGI/edit#gid=0](https://docs.google.com/spreadsheets/d/1RGKj8vVPqQ-PH0UwMX_e9TAzqaYavKi0z0D4pKY9RGI/edit#gid=0) please indicate what you are working on or if a given example was finished

- no mark: to be done
- x: currently working on it
- xx: done

Initials	Full name
CB	Christian Bleiler
AH	Andreas Hessenthaler
TK	Thomas Klotz
AK	Aaron Krämer
BM	Benjamin Maier
SM	Sergio Morales
MM	Mylena Mordhorst
HS	Harry Saini

**Table 1:** Initials of people working on examples, in alphabetical order (surnames).

### 3 DIFFUSION EQUATION

#### 3.1 Equation in general form

The governing equation is,

$$\partial_t u + \nabla \cdot [\sigma \nabla u] = f, \quad (1)$$

with conductivity tensor  $\sigma$ . The conductivity tensor is,

- defined in material coordinates (fibre direction),
- diagonal,
- defined per element.

### 3.2 Example-0001

Example uses generated regular meshes and solves a static problem, i.e., applies the boundary conditions in one step.

#### 3.2.1 Mathematical model - 2D

We solve the following scalar equation,

$$\nabla \cdot \nabla u = 0 \quad \Omega = [0, 2] \times [0, 1], \quad (2)$$

with boundary conditions

$$u = 0 \quad x = y = 0, \quad (3)$$

$$u = 1 \quad x = 2, y = 1. \quad (4)$$

No material parameters to specify.

#### 3.2.2 Mathematical model - 3D

We solve the following scalar equation,

$$\nabla \cdot \nabla u = 0 \quad \Omega = [0, 2] \times [0, 1] \times [0, 1], \quad (5)$$

with boundary conditions

$$u = 0 \quad x = y = z = 0, \quad (6)$$

$$u = 1 \quad x = 2, y = z = 1. \quad (7)$$

No material parameters to specify.

#### 3.2.3 Computational model

- Commandline arguments are:
  - float: length along x-direction
  - float: length along y-direction
  - float: length along z-direction (set to zero for 2D)
  - integer: number of elements in x-direction
  - integer: number of elements in y-direction
  - integer: number of elements in z-direction (set to zero for 2D)
  - integer: interpolation order (1: linear; 2: quadratic)
  - integer: solver type (0: direct; 1: iterative)
- Commandline arguments for tests are:
  - 2.0 1.0 0.0 2 1 0 1 0
  - 2.0 1.0 0.0 4 2 0 1 0
  - 2.0 1.0 0.0 8 4 0 1 0
  - 2.0 1.0 0.0 2 1 0 2 0
  - 2.0 1.0 0.0 4 2 0 2 0
  - 2.0 1.0 0.0 8 4 0 2 0
  - 2.0 1.0 0.0 2 1 0 1 1
  - 2.0 1.0 0.0 4 2 0 1 1

```

2.0 1.0 0.0 8 4 0 1 1
2.0 1.0 0.0 2 1 0 2 1
2.0 1.0 0.0 4 2 0 2 1
2.0 1.0 0.0 8 4 0 2 1
2.0 1.0 1.0 2 1 1 1 0
2.0 1.0 1.0 4 2 2 1 0
2.0 1.0 1.0 8 4 4 1 0
2.0 1.0 1.0 2 1 1 2 0
2.0 1.0 1.0 4 2 2 2 0
2.0 1.0 1.0 8 4 4 2 0
2.0 1.0 1.0 2 1 1 1 1
2.0 1.0 1.0 4 2 2 1 1
2.0 1.0 1.0 8 4 4 1 1
2.0 1.0 1.0 2 1 1 2 1
2.0 1.0 1.0 4 2 2 2 1
2.0 1.0 1.0 8 4 4 2 1

```

### 3.2.4 Result summary

We use CHeart rev. 6292 to produce numerical reference solutions.

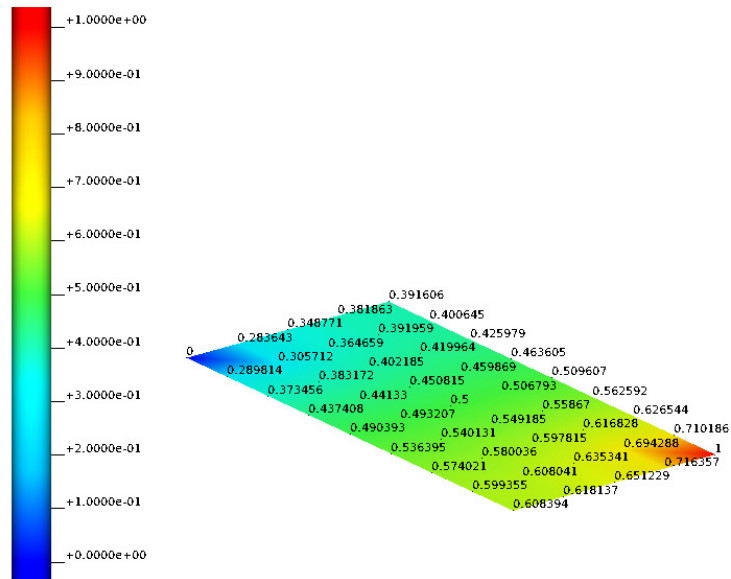


Figure 1: 2D results, iron reference w/ command line arguments [2.0 1.0 0.0 8 4 0 1 o].



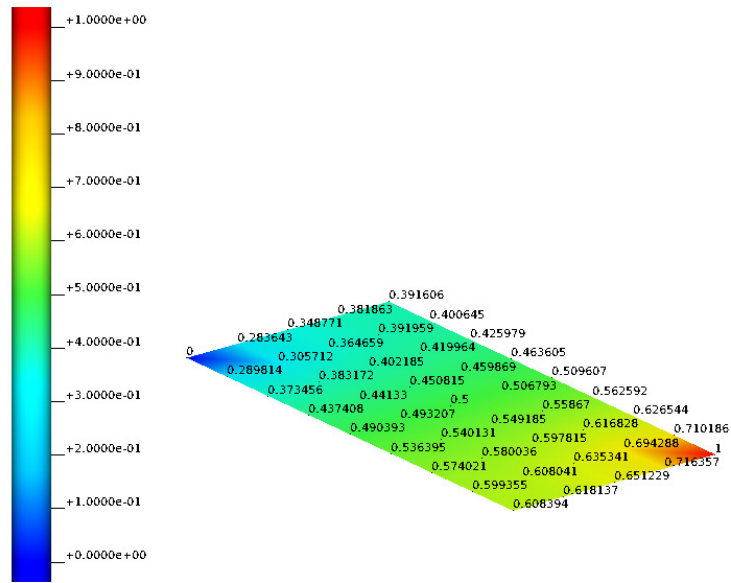


Figure 2: 2D results, current run w/ command line arguments [2.0 1.0 0.0 8 4 0 1 0].

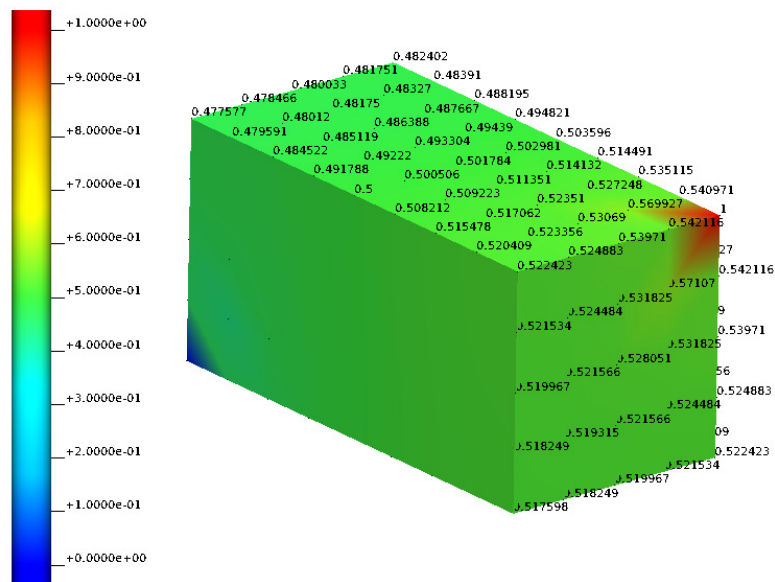


Figure 3: 3D results, iron reference w/ command line arguments [2.0 1.0 1.0 8 4 4 1 0].



### 3.3 Example-0001-u

Example uses user-defined regular meshes in CHeart mesh format and solves a static problem, i.e., applies the boundary conditions in one step.

#### 3.3.1 Mathematical model - 2D

We solve the following scalar equation,

$$\nabla \cdot \nabla u = 0 \quad \Omega = [0, 2] \times [0, 1], \quad (8)$$

with boundary conditions

$$u = 0 \quad x = y = 0, \quad (9)$$

$$u = 1 \quad x = 2, y = 1. \quad (10)$$

No material parameters to specify.

#### 3.3.2 Mathematical model - 3D

We solve the following scalar equation,

$$\nabla \cdot \nabla u = 0 \quad \Omega = [0, 2] \times [0, 1] \times [0, 1], \quad (11)$$

with boundary conditions

$$u = 0 \quad x = y = z = 0, \quad (12)$$

$$u = 1 \quad x = 2, y = z = 1. \quad (13)$$

No material parameters to specify.

#### 3.3.3 Computational model

- Commandline arguments are:
  - float: length along x-direction
  - float: length along y-direction
  - float: length along z-direction (set to zero for 2D)
  - integer: number of elements in x-direction
  - integer: number of elements in y-direction
  - integer: number of elements in z-direction (set to zero for 2D)
  - integer: interpolation order (1: linear; 2: quadratic)
  - integer: solver type (0: direct; 1: iterative)
- Commandline arguments for tests are:
  - 2.0 1.0 0.0 2 1 0 1 0
  - 2.0 1.0 0.0 4 2 0 1 0
  - 2.0 1.0 0.0 8 4 0 1 0
  - 2.0 1.0 0.0 2 1 0 2 0
  - 2.0 1.0 0.0 4 2 0 2 0
  - 2.0 1.0 0.0 8 4 0 2 0
  - 2.0 1.0 0.0 2 1 0 1 1
  - 2.0 1.0 0.0 4 2 0 1 1

```

2.0 1.0 0.0 8 4 0 1 1
2.0 1.0 0.0 2 1 0 2 1
2.0 1.0 0.0 4 2 0 2 1
2.0 1.0 0.0 8 4 0 2 1
2.0 1.0 1.0 2 1 1 1 0
2.0 1.0 1.0 4 2 2 1 0
2.0 1.0 1.0 8 4 4 1 0
2.0 1.0 1.0 2 1 1 2 0
2.0 1.0 1.0 4 2 2 2 0
2.0 1.0 1.0 8 4 4 2 0
2.0 1.0 1.0 2 1 1 1 1
2.0 1.0 1.0 4 2 2 1 1
2.0 1.0 1.0 8 4 4 1 1
2.0 1.0 1.0 2 1 1 2 1
2.0 1.0 1.0 4 2 2 2 1
2.0 1.0 1.0 8 4 4 2 1

```

- Note: Binary uses command line arguments to search for the relevant mesh files.

### 3.3.4 Result summary

We use CHeart rev. 6292 to produce numerical reference solutions.

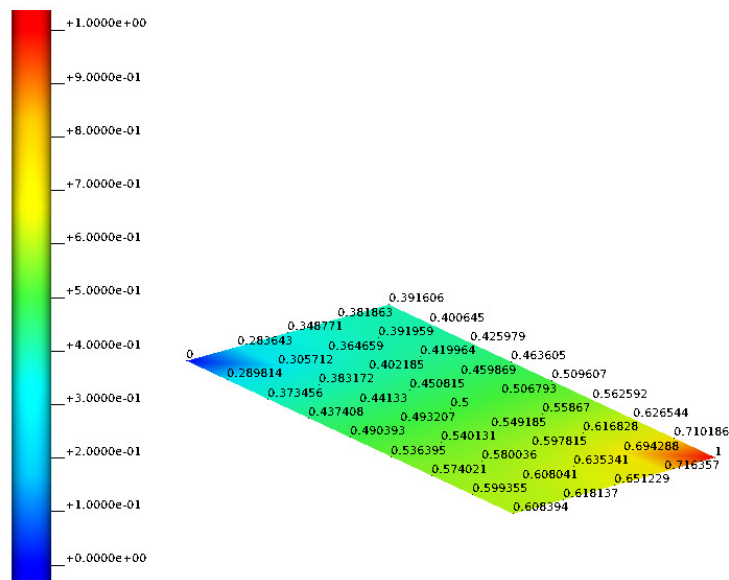


Figure 5: 2D results, iron reference w/ command line arguments [2.0 1.0 0.0 8 4 0 1 o].

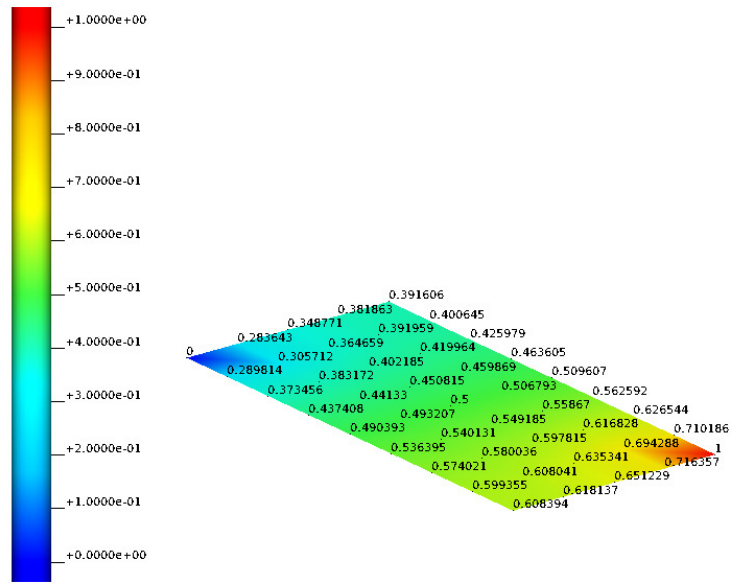


Figure 6: 2D results, current run w/ command line arguments [2.0 1.0 0.0 8 4 0 1 0].

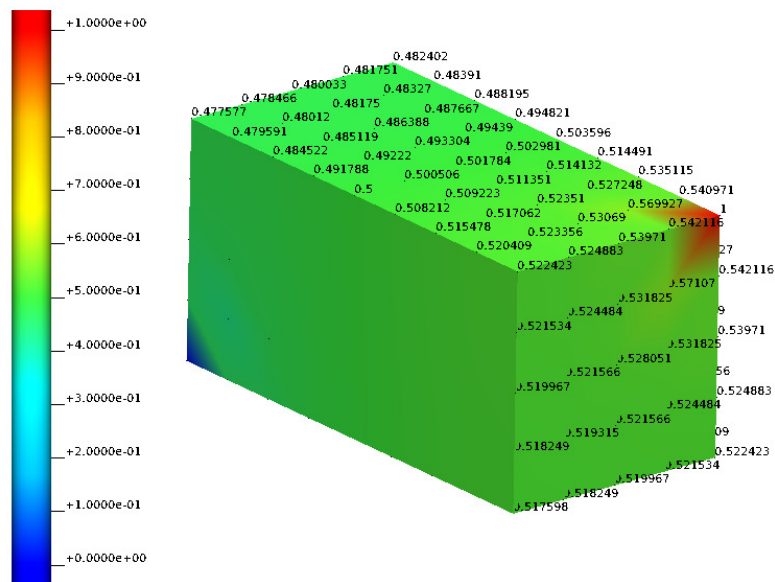


Figure 7: 3D results, iron reference w/ command line arguments [2.0 1.0 1.0 8 4 4 1 0].



### 3.4 Example-0002

Example uses generated regular meshes and solves a static problem, i.e., applies the boundary conditions in one step.

#### 3.4.1 Mathematical model - 2D

We solve the following scalar equation,

$$\nabla \cdot \nabla u = 0 \quad \Omega = [0, 2] \times [0, 1], \quad (14)$$

with boundary conditions

$$u = 15y \quad x = 0, \quad (15)$$

$$u = 25 - 18y \quad x = 2. \quad (16)$$

No material parameters to specify.

#### 3.4.2 Mathematical model - 3D

We solve the following scalar equation,

$$\nabla \cdot \nabla u = 0 \quad \Omega = [0, 2] \times [0, 1] \times [0, 1], \quad (17)$$

with boundary conditions

$$u = 15y \quad x = 0, \quad (18)$$

$$u = 25 - 18y \quad x = 2. \quad (19)$$

No material parameters to specify.

#### 3.4.3 Computational model

- Commandline arguments are:
  - float: length along x-direction
  - float: length along y-direction
  - float: length along z-direction (set to zero for 2D)
  - integer: number of elements in x-direction
  - integer: number of elements in y-direction
  - integer: number of elements in z-direction (set to zero for 2D)
  - integer: interpolation order (1: linear; 2: quadratic)
  - integer: solver type (0: direct; 1: iterative)
- Commandline arguments for tests are:
  - 2.0 1.0 0.0 2 1 0 1 0
  - 2.0 1.0 0.0 4 2 0 1 0
  - 2.0 1.0 0.0 8 4 0 1 0
  - 2.0 1.0 0.0 2 1 0 2 0
  - 2.0 1.0 0.0 4 2 0 2 0
  - 2.0 1.0 0.0 8 4 0 2 0
  - 2.0 1.0 0.0 2 1 0 1 1
  - 2.0 1.0 0.0 4 2 0 1 1

```

2.0 1.0 0.0 8 4 0 1 1
2.0 1.0 0.0 2 1 0 2 1
2.0 1.0 0.0 4 2 0 2 1
2.0 1.0 0.0 8 4 0 2 1
2.0 1.0 1.0 2 1 1 1 0
2.0 1.0 1.0 4 2 2 1 0
2.0 1.0 1.0 8 4 4 1 0
2.0 1.0 1.0 2 1 1 2 0
2.0 1.0 1.0 4 2 2 2 0
2.0 1.0 1.0 8 4 4 2 0
2.0 1.0 1.0 2 1 1 1 1
2.0 1.0 1.0 4 2 2 1 1
2.0 1.0 1.0 8 4 4 1 1
2.0 1.0 1.0 2 1 1 2 1
2.0 1.0 1.0 4 2 2 2 1
2.0 1.0 1.0 8 4 4 2 1

```

### 3.4.4 Result summary

We use CHeart rev. 6292 to produce numerical reference solutions.

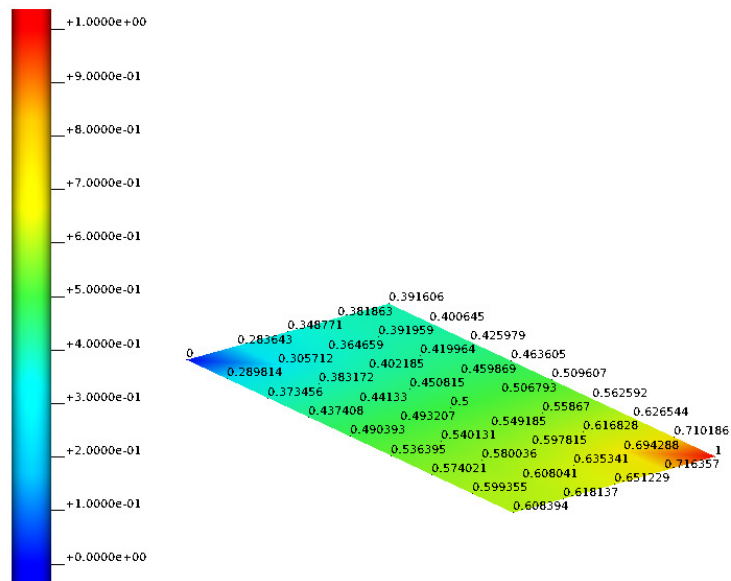


Figure 9: 2D results, iron reference w/ command line arguments [2.0 1.0 0.0 8 4 0 1 o].



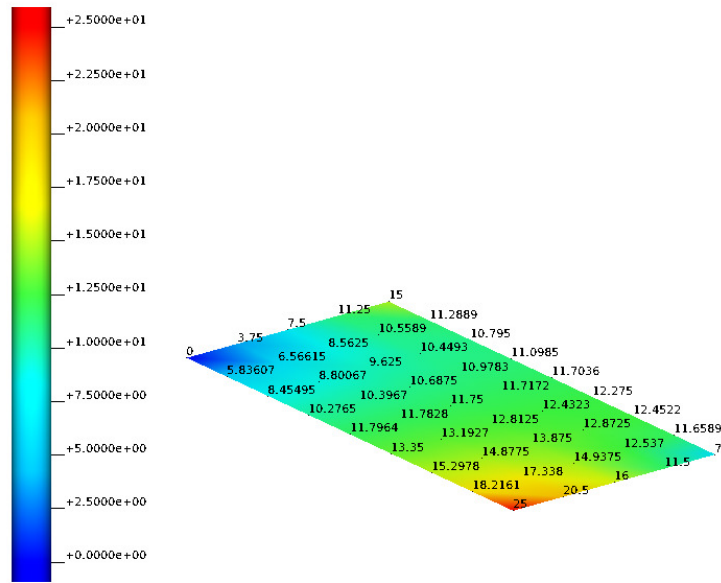


Figure 10: 2D results, current run w/ command line arguments [2.0 1.0 0.0 8 4 0 1 0].

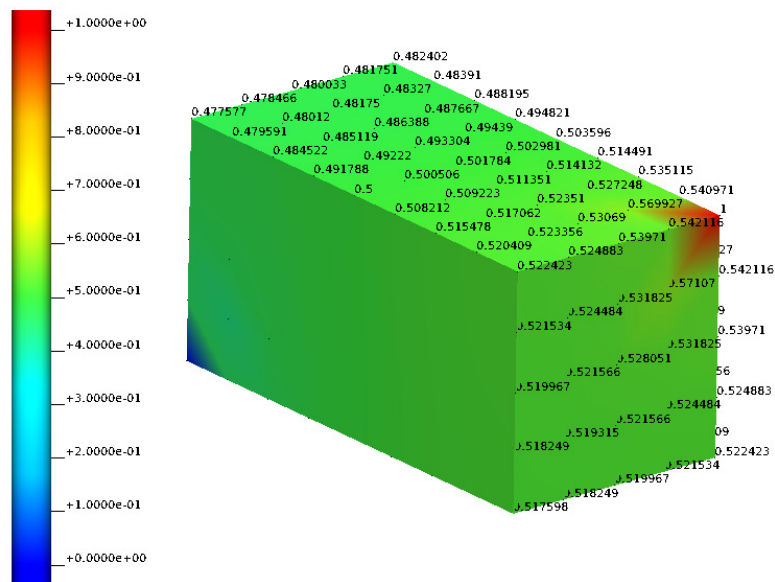


Figure 11: 3D results, iron reference w/ command line arguments [2.0 1.0 1.0 8 4 1 0].

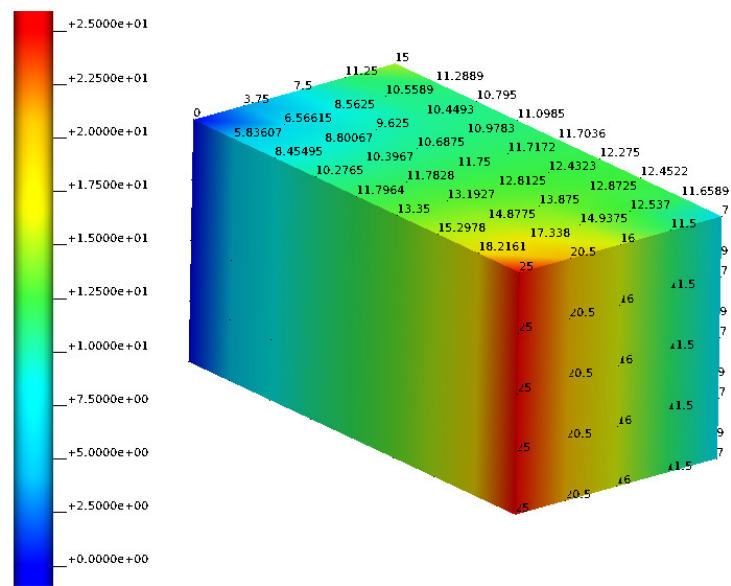


Figure 12: 3D results, current run w/ command line arguments [2.0 1.0 1.0 8 4 4 1 0].

### 3.5 Example-0011

Example uses generated regular meshes and solves a static problem, i.e., applies the boundary conditions in one step.

#### 3.5.1 Mathematical model - 2D

We solve the following scalar equation,

$$\nabla \cdot [\sigma \nabla u] = 0 \quad \Omega = [0, 2] \times [0, 1], \quad (20)$$

with boundary conditions

$$u = 0 \quad x = y = 0, \quad (21)$$

$$u = 1 \quad x = 2, y = 1. \quad (22)$$

The conductivity tensor is defined as,

$$\sigma(x, t) = \sigma = \mathbf{I}. \quad (23)$$

#### 3.5.2 Mathematical model - 3D

We solve the following scalar equation,

$$\nabla \cdot [\sigma \nabla u] = 0 \quad \Omega = [0, 2] \times [0, 1] \times [0, 1], \quad (24)$$

with boundary conditions

$$u = 0 \quad x = y = z = 0, \quad (25)$$

$$u = 1 \quad x = 2, y = z = 1. \quad (26)$$

The conductivity tensor is defined as,

$$\sigma(x, t) = \sigma = \mathbf{I}. \quad (27)$$

#### 3.5.3 Computational model

- Commandline arguments are:
  - float: length along x-direction
  - float: length along y-direction
  - float: length along z-direction (set to zero for 2D)
  - integer: number of elements in x-direction
  - integer: number of elements in y-direction
  - integer: number of elements in z-direction (set to zero for 2D)
  - integer: interpolation order (1: linear; 2: quadratic)
  - integer: solver type (0: direct; 1: iterative)
  - float:  $\sigma_{11}$
  - float:  $\sigma_{22}$
  - float:  $\sigma_{33}$  (ignored for 2D)

- Commandline arguments for tests are:

```

2.0 1.0 0.0 2 1 0 1 0 1 1
2.0 1.0 0.0 4 2 0 1 0 1 1
2.0 1.0 0.0 8 4 0 1 0 1 1
2.0 1.0 0.0 2 1 0 2 0 1 1
2.0 1.0 0.0 4 2 0 2 0 1 1
2.0 1.0 0.0 8 4 0 2 0 1 1
2.0 1.0 0.0 2 1 0 1 1 1 1
2.0 1.0 0.0 4 2 0 1 1 1 1
2.0 1.0 0.0 8 4 0 1 1 1 1
2.0 1.0 0.0 2 1 0 2 1 1 1
2.0 1.0 0.0 4 2 0 2 1 1 1
2.0 1.0 0.0 8 4 0 2 1 1 1
2.0 1.0 1.0 2 1 1 1 0 1 1 1
2.0 1.0 1.0 4 2 2 1 0 1 1 1
2.0 1.0 1.0 8 4 4 1 0 1 1 1
2.0 1.0 1.0 2 1 1 2 0 1 1 1
2.0 1.0 1.0 4 2 2 2 0 1 1 1
2.0 1.0 1.0 8 4 4 2 0 1 1 1
2.0 1.0 1.0 2 1 1 1 1 1 1 1
2.0 1.0 1.0 4 2 2 1 1 1 1 1
2.0 1.0 1.0 8 4 4 1 1 1 1 1
2.0 1.0 1.0 2 1 1 2 1 1 1 1
2.0 1.0 1.0 4 2 2 2 1 1 1 1
2.0 1.0 1.0 8 4 4 2 1 1 1 1

```

### 3.5.4 *Result summary*

We use CHeart rev. 6292 to produce numerical reference solutions.

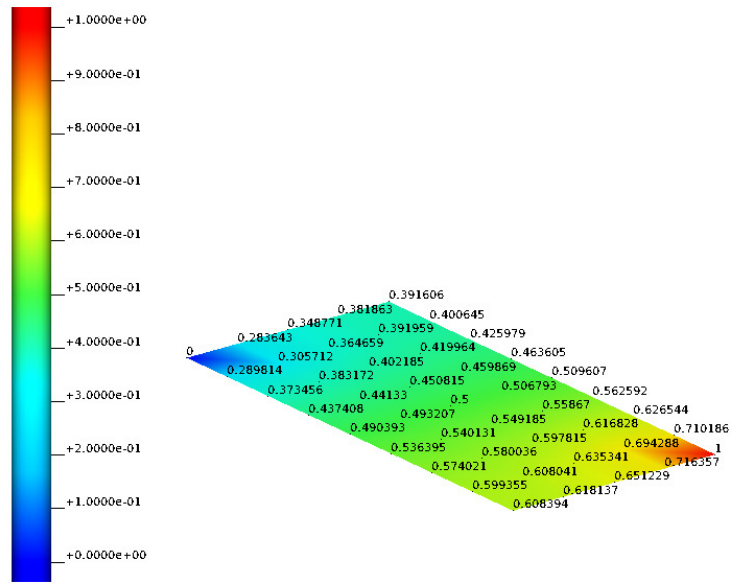


Figure 13: 2D results, iron reference w/ command line arguments [2.0 1.0 0.0 8 4 0 1 0 1 1].

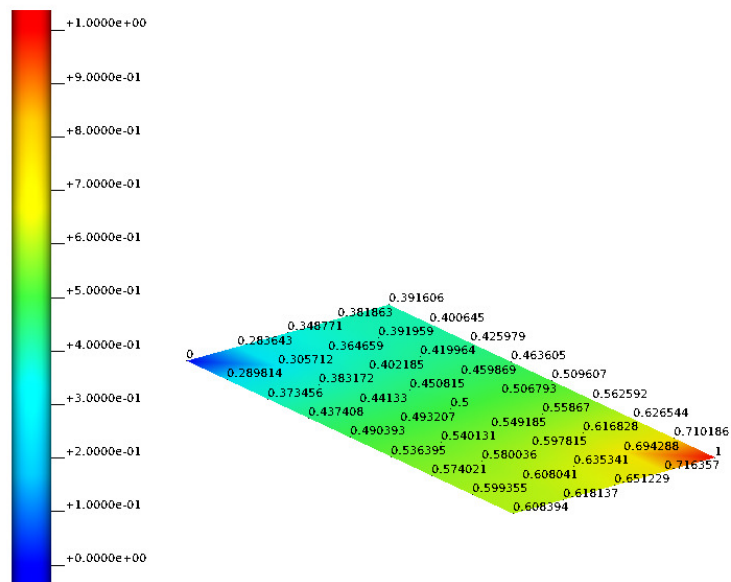


Figure 14: 2D results, current run w/ command line arguments [2.0 1.0 0.0 8 4 0 1 0 1 0].

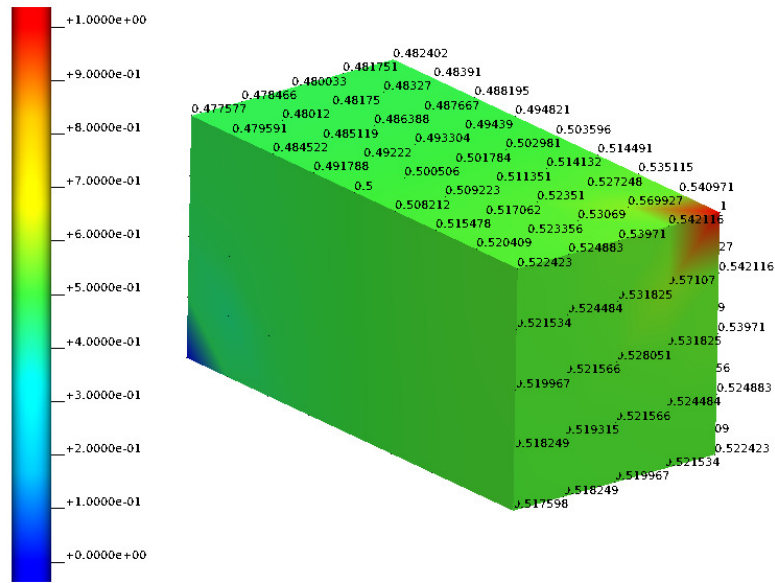


Figure 15: 3D results, iron reference w/ command line arguments [2.0 1.0 1.0 8 4 4 1 0 1 1 1].

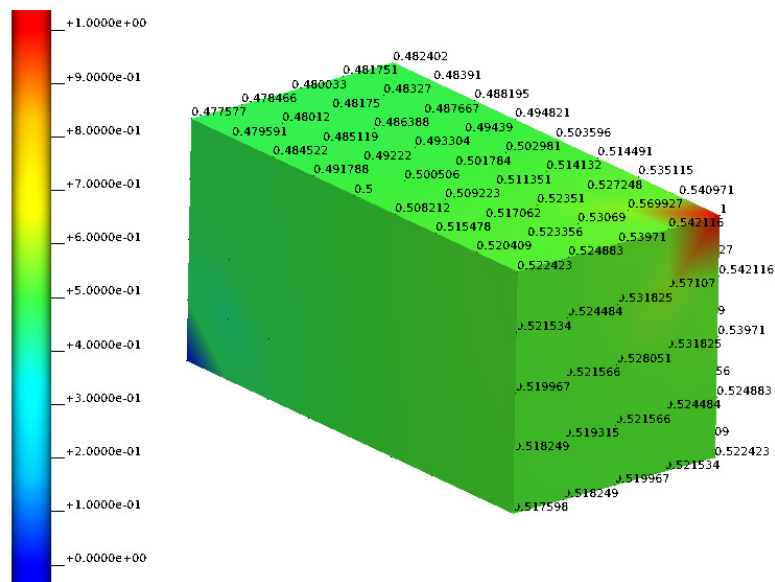


Figure 16: 3D results, current run w/ command line arguments [2.0 1.0 1.0 8 4 4 1 0 1 1 1].

## 4 LINEAR ELASTICITY

### 4.1 Equation in general form

$$\partial_{tt}\mathbf{u} + \nabla \cdot \boldsymbol{\sigma}(\mathbf{u}, t) = \mathbf{f}(\mathbf{u}, t) \quad (28)$$

## 4.2 Example-0101

### 4.2.1 Mathematical model

We solve the following equation,

$$\nabla \cdot \boldsymbol{\sigma}(\mathbf{u}, t) = \mathbf{0} \quad \Omega = [0, 160] \times [0, 120], t \in [0, 5], \quad (29)$$

with time step size  $\Delta_t = 1$  and boundary conditions

$$\dots \quad \dots, \quad (30)$$

$$\dots \quad \dots \quad (31)$$

2D: specify thickness, Young's modulus and Poisson's ratio.

### 4.2.2 Computational model

- Length, width, height
- Direct/iterative solver
- Generated/user mesh
- Number of elements
- Interpolation order
- Number of solver steps (time steps, load steps)

### 4.2.3 Results

Figure 17: Results, analytical solution.

### 4.2.4 Validation

CHeart rev. 6328, Abaqus 2017, analytical reference solution, whatever...



**Figure 18:** Results, Abaqus reference.

**Figure 19:** Results, iron reference.

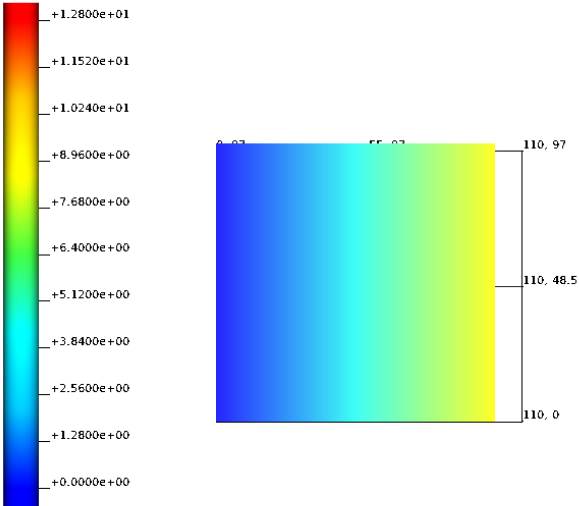


Figure 20: Results, current run.

## 5 FINITE ELASTICITY

## 6 NAVIER-STOKES FLOW

## 7 MONODOMAIN

## 8 CELLML MODEL

## REFERENCES

- [1] Chris Bradley, Andy Bowery, Randall Britten, Vincent Budelmann, Oscar Camara, Richard Christie, Andrew Cookson, Alejandro F Frangi, Thiranjia Babarenda Gamage, Thomas Heidlauf, et al. Openmiss: a multi-physics & multi-scale computational infrastructure for the vph/-physiome project. *Progress in biophysics and molecular biology*, 107(1):32–47, 2011.